

Please replace the paragraph beginning on page 5, line 18 with the following rewritten paragraph:

A3
The portal system may also be configured to conduct searches on behalf of a user. The portal system provides the ability to search both structured data (databases, XML, formatted text, etc.) and unstructured data (HTML files, web-based content, PDF files, etc.) at locations inside and outside the portal. The portal system 120 also allows the user to configure the searches so that only certain objects, in certain locations are searched. By using these search parameters, a user can streamline a search to identify only highly relevant data. This increases the efficiency of the search and reduces the likelihood of identifying undesired results. If a user constructs a search that produces particularly relevant results, then the user may save those search parameters as a channel. The user can return to this channel at a later date to conduct the same search to see if any new objects have been identified. A list of channels stored by a user may be included in a user's portal page, allowing him access to search results by simply clicking on the appropriate channel link.

Please replace the paragraph beginning on page 8, line 11 with the following rewritten paragraph:

A4
cm.t
--FIGURE 1 depicts a high-level illustration of one embodiment of the portal system 120. In FIGURE 1, a plurality of users 100 are connected to a network interface 105 through a computer network 110. The computer network 110 can take many forms including a direct connection, a local-area network, an enterprise intranet, a wireless network, the Internet, or any combination thereof. The network interface 105 is connected to a portal system 120 through a web client 115. Within the portal system 120 are a service broker 125 that controls access to

A4
cancel

the computer system and a plurality of service agents 130 that are configured to perform specific tasks within the portal system 120. Also connected to the portal system 120 are several back-end databases 135, 140, 145, 150 in which data is stored. It should be noted that FIGURE 1 is a block diagram that represents certain functional aspects of the invention as separate blocks. These functional blocks may be implemented on separate computer platforms or on the same computer platform.

Please replace the paragraph beginning on page 10, line 15 with the following rewritten paragraph:

A5
cancel

The service broker 125 serves two functions in the portal system. It controls access to the portal system 120 by users 100 and controls the disposition of jobs to the service agents within the portal system. By controlling the disposition of jobs, the service broker 125 ensures that jobs are processed in an orderly manner and that none of the service agents become overloaded. The event server 215 schedules events, such as jobs, for processing in the portal system 120 on a predefined timetable. The authentication server 220 is used to determine if a particular user should be granted access to the portal system 120. The permissions and group memberships for a particular user are also stored in the authentication server 220. The name server 225 is the storage location for configuration information about all of the other service agents. For example, if the service broker 125 needs to know the location of a specific job server 230, then the name server 225 will provide that information to the service broker 125. The job server 230 is used to execute jobs in the portal system 120. In addition, the job server 230 can retrieve data from a back-end database 200, 205 or 210 to be processed for a particular job. Each job server 230 may be connected to at least one back-end database 200, 205 or 210 in order to

A5
cm

retrieve data therefrom. The job server 230 may also be a stand-alone unit which process jobs that does not retrieve data from any external sources. The repository 235 is used as a storage device for all information that is to be stored in the portal system. All computer files that are stored in the repository 235 are called objects. These objects may include HTML files, job output reports, executable job files (SQL, etc.), image files, etc. Objects that are stored in the repository 235 are arranged in a hierarchy called categories. Within each category, both objects and subcategories may be stored. Categories are thus organized in a tree system much like the file system on a standard computer. In addition, each object in the repository may include more than one version. Versioning can be used to accomplish a variety of objectives including setting multiple security levels for different versions of an object, and allowing a user to see a modification history of an object. The knowledge server 250 provides the search and channel functions for the portal system 120. The knowledge server 250 is comprised of two components: a search server 245 and a crawl server 250. The crawl server 250 uses one or more crawlers to analyze and index specific information that is stored in the repository 235, a company intranet, or the Internet. A crawler can be configured to search only in certain locations in the repository 235, a company intranet, or the Internet for information to be indexed. The indices produced by the crawl server 245 are stored in the knowledge server 240 in files called information sources. Depending upon the settings of the crawl server 250, an information source will contain an index of objects found both within the portal system (i.e. in the repository 235), or outside the portal system (i.e. on an intranet or the Internet). The crawl server 250 is capable of indexing structured and unstructured data. The search server 245 uses the information sources produced by the crawl server 250 to conduct searches on behalf of a user. Because the information sources will generally correspond to specialized topics, a user may increase the efficiency of a search by

A5
Cmcd
selecting only those information sources that are relevant to his search. The portal system 120 can include redundant service agents for processing user requests. In this manner, the portal system 120 is scalable to handle both a small enterprise with a small number of users and a large enterprise with many redundant service agents for processing requests from thousands of users.--

Please replace the paragraph beginning on page 31, line 17 with the following rewritten paragraph:

A6
Cm.t
FIGURE 6 illustrates some of the common properties associated with service agents in the portal system 120. In FIGURE 6, a list of installed service agents 600 is maintained in the Relational Database in the portal system 120. Several properties are associated with each of the installed service agents, including the Name of the service agent 605, the Type of the service agent 610, the Host where the service agent resides 615, the Database Type 620 associated with service agent, and the Database Server 625 associated with the service agent. As seen in FIGURE 6, a list of the different service agent types 630 is maintained in the Relational Database in the portal system 120. Thus, when a service agent is installed on the portal system 120, the administrator must add a new service agent type or select from the list of available service agent Types 630. Also depicted in FIGURE 6 is a list of the available service agent Hosts 635 that are resident on the portal system 120. Again, when a new service agent is installed on the portal system 120, the administrator must add a new service agent Host or select from the list of available service agent Hosts 635. More than one service agent type can be installed on a particular service agent host. For example, an authentication server 220 and a name server 225 may both be installed the same host computer. FIGURE 6 also illustrates that each service agent will be assigned a Database Type 620 and a Database Server 625. A list of

AG
cancel the available Database Types 640 and Database Server Types 650 are maintained in the Relational Database in the portal system 120. Generally, only one Database Type 620 will be associated with a Database Server 625.

Please replace the paragraph beginning on page 33, line 11 with the following rewritten paragraph:

AT The search server 245 is assigned certain properties that are illustrated in FIGURE 8. As with all other service agents, the search server 245 is assigned a name, host, database type, and database server. In addition, each job server 230 is also assigned Search Engine Properties 800 and Information Sources 805. The Search Engine Properties 800 describe the operating parameters for the search engine, including a Query Port value, an Index Port value, a Language, an Index Hyphenated Words flag, and Hyphen Character. The Query Port value identifies that port which will handle simultaneous queries from a user. The Index Port value identifies the port that receives indexing requests from the crawl server. The Language specifies the language in which month names and abbreviations of month names appear. The Index Hyphenated Words flag indicates whether the Search Engine indexes a hyphenated word as well as individual words. The Information Sources properties 805 describe the names and properties of each Information Source that has been indexed by the crawl server. Some of the properties include a Name, a Description, a Number of Documents that may be stored in the Information Source, an Expiration Date, and an Expiration Action for the Information Source.

Please replace the paragraph beginning on page 35, line 1 with the following rewritten paragraph:

A8 The General Properties 900 define the basic parameters to be used by each crawler in conducting its search. Included in these parameters are the name of the crawler, the information sources to be searched by the crawler, a list of the URLs to crawl, a flag indicating whether the URLs are case sensitive, and a flag indicating whether the crawler should follow links to other sites.

Please replace the paragraph beginning on page 35, line 6 with the following rewritten paragraph:

A9 The Schedule Properties 905 are used to define the schedule on which the crawler operates. Included within the Schedule Properties 905 are a schedule (as disclosed in figure 5), a start time, and a repeat interval.

Please replace the paragraph beginning on page 35, line 9 with the following rewritten paragraph:

A10 The Limits 910 are used to prevent the crawler from infinitely indexing sites that have deep links. Some of the limits which may be defined for a crawler include a maximum depth (which specifies how many levels of links a crawler may follow), a site duration (which defines the number of hours a crawler should spend on any given site), a page delay (which defines the number of seconds the crawler should wait after downloading a page before moving to the next link of set of links), and a page timeout (which defines the number of seconds that the crawler will wait to receive data after requesting a page).

Please replace the paragraph beginning on page 35, line 16 with the following rewritten paragraph:

•The Directory Properties 915 provide an Allowed field and a Disallowed field.

A11 The allowed field defines a set of strings that must exist in a URL object for that object to be indexed. The disallowed field defines a set of strings that must not be in the URL object in order for the object to be indexed. These fields may be used to greatly limit the number of sites that are indexed in an information source. •

Please replace the paragraph beginning on page 35, line 21 with the following rewritten paragraph:

•Web Pages Properties 920 are used to exclude HTML and text files from being indexed in an information source based upon the content of the document and/or header. Using these properties, a user can specify those strings that must exist and/or those strings that must not exist. If either of these constraints is not met, then the HTML document will not be indexed.

A12 For example, if a user does not want any HTML or text files to be indexed that contain the phrase “living abroad,” the term “living abroad” is added as a Cannot Have Text string. Or, if a user desires to index only all the HTML or text files that contain the phrase “electronic filing,” then the term “electronic filing” would be added as a Must Have Text string. HTML or text files must meet both constraints in order to be indexed in an information source. If a document meets both the Must Have constraint and the Can’t Have constraint, then the Can’t Have constraint takes precedence and the document will not be indexed. Accordingly, the web pages properties will include two fields: a Must Have Text field and a Cannot Have Text field. Other parameters may also be set to look for these strings in the headers or content only. •

Please replace the paragraph beginning on page 37, line 3 with the following rewritten paragraph:

AB Cookies Properties 935 are used to specify additional login information as well as cookies that should be sent to a webserver. The additional login information allows the crawler to log into a site in which a form requires more than just a user name and password. For example, if the form has an input field that specifies a user's age, then age can be specified as additional login information. Therefore, the field's name and value should be added to the cookie as additional login information. Often a form-based login will result in a cookie being submitted to a user's browser interface. The cookie contains information about the user's login parameters insuring that the user and only that user see the content on that website. The cookies properties 935 allows cookie-based login, or cookie spoofing. Using this method, a crawler can sometimes leap frog the login process all together and insure that the crawler appears logged in as soon as it arrives at the site. Once the names and values of the cookies have been determined, they may be added to the cookies page of the crawler, thus allowing the crawler to login when it starts. Of course, if the cookie expires then cookie-based authentication will work for only the duration that the cookie is valid.

Please replace the paragraph beginning on page 39, line 19 with the following rewritten paragraph:

AI4 Cmt --FIGURE 10 depicts a representative example of a portal page 1000 as seen from a user's browser program. A portal page 1000 is a customized web page that presents data from the portal system 120 that is most relevant to a particular user. A portal page 1000 is a user's primary interface to the data, reports and jobs that are resident in the portal system 120.

Each of the sets of data that are presented to a user at the portal page 1000 is called a portal object. A portal page 1000 presents portal objects to a user 100 in a format that is readable by a standard browser program. The user's default portal page 1000 is the first page he sees after logging into the portal system 120. The first time a user logs into the portal system 120, a portal page 1000 is automatically generated. Thereafter, a user 100 can modify his respective copy of the portal page 1000, and can create additional ones. Users can modify the content, layout, and colors of any of their portal pages 1000, as well as changing which portal page 1000 is the default page (the one that displays at the beginning of a portal session). Users to whom permission is granted can publish their own portal pages for others to copy. Users can add additional components to any of their portal pages 1000, or remove optional components. In addition to using pre-configured content provided by the administrator, a user can create and include on his portal page 1000 other content of interest to him, such as bookmarks, channels, and job output files displayed as portal objects.

Please replace the paragraph beginning on page 41, line 1 with the following rewritten paragraph:

--In FIGURE 10, the Broadcast Messages portal object 1005 is configured as a mandatory portal object. Thus, it cannot be modified or removed by a user 100. Accordingly, the Broadcast Message portal object in FIGURE 10 does not include an "X" icon 1008 in the upper right hand corner which would allow a user to delete this portal object from their portal page 1000. Beneath the Broadcast Message portal object 1005, is the Company Billboard portal object 1010. The Company Billboard portal object 1010 is a preconfigured category that includes a list of links that may be accessed, but not edited by the user 100. By clicking on one

A15
Cmld.

of these links, a corresponding object will be displayed in the display window 1025 on the right-hand side of the portal page 1000. In the example of FIGURE 10, the Personal Dashboard link in the Broadcast Messages category 1005 has been selected by a user; thus causing the Personal Dashboard portal object to be displayed in the display window 1025 of the portal page 1000. Because the Company Billboard 1010 has been set up as a preconfigured category, a user 100 cannot edit the links that are listed in this window. A user 100 does have the option of removing this portal object from his personalized portal page by selecting the "X" icon 1008 in the upper right hand corner of the Company Billboard portal object.--

Please replace the paragraph beginning on page 43, line 1 with the following rewritten paragraph:

A16

→ The portal object illustrated at the bottom of the right-most column 1003 of the portal page 1000 is a syndicated content portal object entitled E-commerce Top Stories 1030. A syndicated content portal object is used to present dynamically updated content provided by a third party. Third party content could include information such as a news-wire, a stock quote service, or a sports score reporting service. In FIGURE 10, the third party content that is provided in this window is a news-wire service related to E-commerce stories.↵

Please replace the paragraph beginning on page 46, line 8 with the following rewritten paragraph:

A17
CmT

→A job may require a fresh set of data to be retrieved from a back-end database 200, 205, or 210. If this is the situation, then the job will be dispatched to a job server 230 that is connected to an appropriate back-end database. After the data is retrieved from a back-end database, it is processed by the job server 230 and an output report is prepared. In many

AM
cncl'd.

instances, the output report will have to be transformed into a format that is appropriate for storage and/or presentation to the user. This process is performed by the job server 230. After the output reports have been converted into the appropriate format, then the job server analyzes the job metadata to determine if there are any subscriptions or notifications that need to be fulfilled. The job server 230 will also test the output report to determine if any exception conditions are present. The job server 230 utilizes the job metadata to determine if any of the exception conditions have been met. If so, appropriate notifications will be sent to the subscribing users via e-mail or portal object update (such as notification on a dynamic portal object). In addition, the job server 230 will assign user and group permissions to the output report. These permissions define the users and groups that can access and view the output report. The output report will then be transmitted to the service broker 125 so that it can be forwarded to the user who requested it. The output report will generally be displayed in the display window 1025 of the user's personalized portal page. A copy of the output report may also be stored in the repository 235.

Please ~~replace~~ the paragraph beginning on page 50, line 20 with the following rewritten paragraph:

Jobs that are placed in the retry queue are retried using the following algorithm.

Each job will be retried until they run successfully or until the event server 235 is stopped.

A18
cncl'd

When a job is retried, it follows the five steps outlined above. When a job is first placed in the retry queue it is set to run 10 minutes after the time it is placed in the queue. After each attempt to run the job, if it fails, it is resubmitted to the retry queue with an additional 10-minute wait time added until the next retry. In other words, jobs submitted to the retry queue will be retried